

```

/*
    Ant Project. Otago Polytechnic, New Zealand. 2009 3rd Year B.I.T. project for
    Jun Cui, Gareth Dorset and Trevor Farquharson
    -:~*._.*~:-

    Client: Otago Museum

    Developers: Jun Cui - 3rd year student Otago Polytechnic
                Gareth Dorset - 3rd year student Otago Polytechnic
                Trevor Farquharson - 3rd year student Otago Polytechnic

    Mentors: Patricia Haden - Otago Polytechnic
            Hamish Smith - Otago Polytechnic
            Sam Mann - Otago Polytechnic
*/

// Setup.
fscommand("fullscreen", "true");           // forces the program to run fullscreen
Mouse.hide();                             // hides the mouse pointer in the
program

// Variable Setup
var sWidth = stage.stageWidth;             // width of screen, passed into
constructors
var sHeight = stage.stageHeight;          // height of screen, passed into constructors
var sFrames = stage.frameRate;            // frame rate of program, passed into
constructors

var fItem = 0;                             // determines which item in the
food array appears next

// set up array of food items and max number of food
var food = new Array();
var nFood = 5;

// set up array of ants and max number of ants
var ants = new Array();
var nAnts = 50;

// set up array of objects and max number of objects
//var objects = new Array();
//var nObjects = 9;

// set up motion detection (person)

// person is the input from the motion detection
var person = new TPerson(sFrames, sWidth, sHeight);
addChild(person);

// create food items and add to array
for(var f = 0; f < nFood; f++) {
    // randomly generate x and y values, keep food in lower half of screen
    // to encourage ants to cover full area
    var nFoodX = 100 + (Math.random() * (sWidth - 200));
    var nFoodY = (sHeight / 2) + (Math.random() * ((sHeight / 2) - 100));

    // create the food
    var foodItem = new TFood(nFoodX, nFoodY, sWidth, sHeight);
    // add food to screen
    addChild(foodItem);
    // add food to array for accessing properties
    food.push(foodItem);
}

// create ants and add to array
for(var a = 0; a < nAnts; a++) {
    // set a random number of cycles (frames) before ant appears
    var rCycles = Math.floor(Math.random() * 2500);
    // create the ant
    var ant = new TAnt(sWidth, sHeight, rCycles);
    // add to screen
    addChild(ant);
    // add to array
    ants.push(ant);
}

// create objects and add to array
//for(var o = 0; o < nObjects; o++) {
//    // randomise the position of the stone, twigs, bark and leaves.
//    // set the x positon no closer than 100 pixels from either edge
//    var nObjectX = 100 + (Math.random() * (sWidth - 200));
//    // set the y positon no closer than 100 pixels from either edge
//    var nObjectY = 100 + (Math.random() * (sHeight - 200));
//    // loops through the objects -> mod by 4 because there is 4 frames to the objects movieclip
//    var nObjectFrame = (o % 4) + 1;
//    // create the object
//    var objectItem = new TObject(nObjectX, nObjectY, nObjectFrame, sWidth, sHeight);
//    // add the object to screen
//    addChild(objectItem);
//}

//
// //add to the objects array

```

```

//      objects.push(objectItem);

//}

// setup and add background border (the fading to black around the edges)
var bg = new TBackground(0, 0);
addChild(bg);

// Set up listeners
stage.addEventListener(Event.ENTER_FRAME, main);

// Main Function
function main(e: Event) {
    // Local variables
    var makeFoodVisible: Boolean = true;

    // decrement counter in objects, after about 2 hours the objects will change location
    // designed like this so the area doesn't look static and boring, potentially different
    // everytime a person walks past
    //for(var o = 0; o < nObjects; o++) {
    //      objects[o].DecrementCounter();
    // }

    // process ant behaviours
    for (var a = 0; a < nAnts; a++) {
        // get the person's location
        var pX = person.GetXLoc();
        var pY = person.GetYLoc();

        // enable for following the mouse
        //var pX = mouseX;
        //var pY = mouseY;

        // process behaviour transition method (action gets called within this method)
        ants[a].Transition(ants, food, pX, pY);
    }

    // process food trails and if necessary, make food appear
    for (var f = 0; f < nFood; f++) {
        // decrement counter in food, same function as in objects except every 1 hour,
        // so ants aren't always picking up food from same place
        food[f].DecrementCounter();

        if (food[f].GetCurrFrame() != 42) {
            // if a food item is already on screen (frame is not 42),
            // then don't make any more food appear
            makeFoodVisible = false;
        }

        // if the food has a trail associated with it, decrease the strength of the trail
        if (food[f].GetHasPath()) {
            var tempStr = food[f].GetPathStr();
            if (tempStr >= 5) {
                food[f].SetPathStr(tempStr - 1);
            } else {
                // food trail is too weak, so remove it completely
                food[f].ResetPath();
            }
        }
    }

    // if there was no food visible on screen, then make a new one appear
    if (makeFoodVisible) {
        // ...but only if the random number is less than or equal to 1
        if (Math.floor(Math.random() * 1000) <= 1) {
            food[fItem].ChangeFrame(1);
            // cycle through the food items
            fItem = (fItem + 1) % nFood;
        }
    }
}

```